Simple **garbage-collector**-safety
... doi.acm.org/10.1145/231379.231394 **What is a** DOI? ... to existing restrictions in the language **definiti n**. ... that under minimal, clearly **defined** assumptions about the ...
portal.acm.org/citation.cfm?id=231394 - Similar pages
[More results from portal.acm.org]

[PDF] Very-Long **Instruction** Word (VLIW) Computer Architecture
File Format: PDF/Adobe Acrobat - View as HTML
... has an add **instructi n** that can **enc de** a **memory** ... to discover maximal amounts of **instruction**-level parallelism. Higher degrees of **superscalar** execution, ie, more ...
www.semiconductors.philips.com/ acrobat/other/vliw-wp.pdf - Similar pages

Performance of a hardware-assisted real-time garbage collector

Full text Pdf (1.16 MB)

Source **Architectural Support for Programming Languages and Operating Systems** archive
Proceedings of the sixth international conference on Architectural support for programming languages and operating systems table of contents
San Jose, California, United States
Pages: 76 - 85
Year of Publication: 1994
ISBN:0-89791-660-3
Also published in ...

Authors William J. Schmidt IBM Corporation, 3605 Hwy 52 North, 4B4/005-2, Rochester, MN
Kelvin D. Nilsen Department of Computer Science, Iowa State University, Ames, IA

Sponsors SIGOPS: ACM Special Interest Group on Operating Systems
IEEE-CS : Computer Society
SIGARCH: ACM Special Interest Group on Computer Architecture
SIGPLAN: ACM Special Interest Group on Programming Languages

Publisher ACM Press New York, NY, USA

DOI Bookmark: Use this link to bookmark this Article: http://doi.acm.org/10.1145/195473.195504
What is a DOI?

↑ ABSTRACT

Hardware-assisted real-time garbage collection offers high throughput and small worst-case bounds on the times required to allocate dynamic objects and to access the memory contained within previously allocated objects. Whether the proposed technology is cost effective depends on various choices between configuration alternatives. This paper reports the performance of several different configurations of the hardware-assisted real-time garbage collection system subjected to several different workloads. Reported measurements demonstrate that hardware-assisted real-time garbage collection is a viable alternative to traditional explicit memory management techniques, even for low-level languages like C++.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 Andrew W. Appel, Garbage collection can be faster than stack allocation, Information Processing Letters, v.25 n.4, p.275-279, June 17, 1987

2 Henry G. Baker, Jr., List processing in real time on a serial computer, Communications of the ACM, v.21 n.4, p.280-294, April 1978

3 Hans-Juergen Boehm , Mark Weiser, Garbage collection in an uncooperative environment, Software—Practice & Experience, v.18 n.9, p.807-820, September 1988

4 Rodney A. Brooks, Trading data space for reduced time and code space in real-time garbage collection on stock hardware, Proceedings of the 1984 ACM Symposium on LISP and functional programming, p.256-262, August 06-08, 1984, Austin, Texas, United States

5 A. W. Appel , J. R. Ellis , K. Li, Real-time concurrent collection on stock multiprocessors, Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation, p.11-20, June 20-24, 1988, Atlanta, Georgia, United States

6 H. Gao and K. D. Nilsen, Rehable general purpose dynamic memory management for real-time systems, Technical Report 94-09, Department of Computer Science, Iowa State University, 1994.

7 David A. Patterson , John L. Hennessy, Computer architecture: a quantitative approach, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1990

8 Samuel N. Kamin, Programming languages: an interpreter-based approach, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1990

9 David A. Moon, Garbage collection in a large LISP system, Proceedings of the 1984 ACM Symposium on LISP and functional programming, p.235-246, August 06-08, 1984, Austin, Texas, United States

10 David A. Moon, Symbolics architecture, Computer, v.20 n.1, p.43-52, Jan. 1987

11 Motorola, Power PC 601 RISC M~croprocessor User's Manual, 1993.

12 K. Nilsen, Garbage collection of strings and linked data structures in real time, Software—Practice & Experience, v.18 n.7, p.613-640, July 1988

13 Kelvin Nilsen, Reliable real-time garbage collection of C++, Computing Systems, v.7 n.4, p.467-504, Fall 1994

14 Kelvin D. Nilsen , William J. Schmidt, Cost-effective object space management for hardware-assisted real-time garbage collection, ACM Letters on Programming Languages and Systems (LOPLAS), v.1 n.4, p.338-354, Dec. 1992

15 K. D. Nilsen and W. J. Schmidt, A high-performance hardware-assisted real-time garbage collection system, Journal oj: Programming Languages (1994), to appear.

16 William Jon Schmidt, Issues in the design and implementation of a real-time garbage collection architecture, Iowa State University, Ames, IA, 1992

17 M. D. Tiemann, User's Gu,de to GNU C++, Free Software Foundation, 1990, Version 1.37.1. Available by anonymous ~tp from prep.ai.rait, edu.

18 David Michael Ungar, The design and evaluation of a high performance Smalltalk system, MIT Press, Cambridge, MA, 1987

19 David Ungar , Ricki Blau , Peter Foley , Dain Samples , David Patterson, Architecture of SOAR:

Smalltalk on a RISC, Proceedings of the 11th annual international symposium on Computer architecture, p.188-197, January 1984

20 Skef Wholey , Scott E. Fahlman, The design of an instruction set for common LISP, Proceedings of the 1984 ACM Symposium on LISP and functional programming, p.150-158, August 06-08, 1984, Austin, Texas, United States

21 M. Wolczko and I. Williams, The influence of the object-oriented language model on a supporting architecture, 26th Hawaii Ir~ternat~onal Uonfcrencc on System Sciences, January 1993.

↑ CITINGS

Dirk Grunwald , Richard Neves, Whole-program optimization for time and space efficient threads, ACM SIGPLAN Notices, v.31 n.9, p.50-59, Sept. 1996

↑ INDEX TERMS

Primary Classification:

D. Software
- ↳ **D.4** OPERATING SYSTEMS
  - ↳ **D.4.2** Storage Management
    - ↳ **Subjects:** Allocation/deallocation strategies

Additional Classification:

C. Computer Systems Organization
- ↳ **C.3** SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS
  - ↳ **Subjects:** Real-time and embedded systems

D. Software
- ↳ **D.3** PROGRAMMING LANGUAGES
  - ↳ **D.3.4** Processors
    - ↳ **Subjects:** Compilers

General Terms:

Algorithms, Design, Experimentation, Languages, Measurement, Performance, Theory

↑ Collaborative Colleagues:

Kelvin D. Nilsen:	Bernt Rygg
	William J. Schmidt
William J. Schmidt:	Steven R. Kunkel
	Mikko H. Lipasti
	Jack H. Lutz
	Kelvin D. Nilsen
	Robert R. Roediger

↑ Peer t Peer - Readers of this Article have als read:

- Data structures for quadtree approximation and compression

Communicati ns of the ACM 28, 9
Hanan Samet

- A hierarchical single-key-lock access control using the Chinese remainder theorem
  Pr ceedings f the 1992 ACM/SIGAPP Symposium on Applied c mputing
  Kim S. Lee , Huizhu Lu , D. D. Fisher

- The GemStone object database management system
  Communications of the ACM 34, 10
  Paul Butterworth , Allen Otis , Jacob Stein

- Putting innovation to work: adoption strategies for multimedia communication systems
  Communications of the ACM 34, 12
  Ellen Francik , Susan Ehrlich Rudman , Donna Cooper , Stephen Levine

- An intelligent component database for behavioral synthesis
  Proceedings of the 27th ACM/IEEE conference on Design automation
  Gwo-Dong Chen , Daniel D. Gajski

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
  Volume 29 , Issue 11 (November 1994)

- **ACM SIGOPS Operating Systems Review**
  Volume 28 , Issue 5 (December 1994)

Terms used **superscalar instruction garbage** Found **3,700** of **147,060**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Relational profiling: enabling thread-level parallelism in virtual machines
Timothy Heil, James E. Smith
December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**
Full text available: pdf(237.19 KB) ps(1.61 MB) Additional Information: full citation, references, citings, index terms
Publisher Site

2 Exploiting prolific types for memory management and optimizations
Yefim Shuf, Manish Gupta, Rajesh Bordawekar, Jaswinder Pal Singh
January 2002 **ACM SIGPLAN Notices , Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 37 Issue 1
Full text available: pdf(203.59 KB) Additional Information: full citation, abstract, references, citings

In this paper, we introduce the notion of *prolific* and *non-prolific* types, based on the number of instantiated objects of those types. We demonstrate that distinguishing between these types enables a new class of techniques for memory management and data locality, and facilitates the deployment of known techniques. Specifically, we first present a new *type-based* approach to garbage collection that has similar attributes but lower cost than generational collection. Then we de ...

3 Concurrent garbage collection using hardware-assisted profiling
Timothy H. Heil, James E. Smith
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management**, Volume 36 Issue 1
Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, citings, index terms

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

4 Improving Java performance using hardware translation
Ramesh Radhakrishnan, Ravi Bhargava, Lizy K. John

June 2001 **Proceedings f the 15th internati nal c nference n Superc mputing**

Full text available: pdf(254.91 KB) Additional Information: full citation, abstract, references, citings, index terms

State of the art Java Virtual Machines with Just-In-Time (JIT) compilers make use of advanced compiler techniques, run-time profiling and adaptive compilation to improve performance. However, these techniques for alleviating performance bottlenecks are more effective in long running workloads, such as server applications. Short running Java programs, or client workloads, spend a large fraction of their execution time in compilation instead of useful execution when run using JIT compilers. In ...

5 Using complete system simulation to characterize SPECjvm98 benchmarks

Tao Li, Lizy Kurian John, Vijaykrishnan Narayanan, Anand Sivasubramaniam, Jyotsna Sabarinathan, Anupama Murthy

May 2000 **Proceedings of the 14th international conference on Supercomputing**

Full text available: pdf(1.66 MB) Additional Information: full citation, abstract, references, citings, index terms

Complete system simulation to understand the influence of architecture and operating systems on application execution has been identified to be crucial for systems design. While there have been previous attempts at understanding the architectural impact of Java programs, there has been no prior work investigating the operating system (kernel) activity during their executions. This problem is particularly interesting in the context of Java since it is not only the application that can invoke ...

6 Exploiting ILP in page-based intelligent memory

Mark Oskin, Justin Hensley, Diana Keen, Frederic T. Chong, Matthew Farrens, Aneet Chopra

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.35 MB) Publisher Site Additional Information: full citation, abstract, references, index terms

This study compares the speed, area, and power of different implementations of Active Pages [OCS98], an intelligent memory system which helps bridge the growing gap between processor and memory performance by associating simple functions with each page of data. Previous investigations have shown up to 1000X speedups using a block of reconfigurable logic to implement these functions next to each sub-array on a DRAM chip. In this study, we show that instruction-level parallelism, n ...

7 Garbage collection for strongly-typed languages using run-time type reconstruction

Shail Aditya, Christine H. Flood, James E. Hicks

July 1994 **ACM SIGPLAN Lisp Pointers , Proceedings of the 1994 ACM conference on LISP and functional programming**, Volume VII Issue 3

Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Garbage collectors perform two functions: live-object detection and dead-object reclamation. In this paper, we present a new technique for live-object detection based on run-time type reconstruction for a strongly typed, polymorphic language. This scheme uses compile-time type information together with the run-time tree of activation frames to determine the exact type of every object participating in the computation. These reconstructed types are then used ...

8 Fast out-of-order processor simulation using memoization

Eric Schnarr, James R. Larus

October 1998 **Pr ceedings f the eighth international c nference n Architectural supp rt f r pr gramming languages and perating systems**, Volume 32 , 33

Issue 5 , 11

Full text available: pdf(1.43 MB) Additional Information: full citation, abstract, references, citings, index terms

Our new out-of-order processor simulatol; FastSim, uses two innovations to speed up simulation 8--15 times (vs. Wisconsin SimpleScalar) with no loss in simulation accuracy. First, FastSim uses speculative direct-execution to accelerate the functional emulation of speculatively executed program code. Second, it uses a variation on memoization---a well-known technique in programming language implementation---to cache microarchitecture states and the resulting simulator actions, and then "fast forw ...

Keywords: direct-execution, memoization, out-of-order processor simulation

9 The cache behaviour of large lazy functional programs on stock hardware

Nicholas Nethercote, Alan Mycroft

June 2002 **ACM SIGPLAN Notices , Proceedings of the workshop on Memory system performance**, Volume 38 Issue 2 supplement

Full text available: pdf(1.26 MB) Additional Information: full citation, abstract, references, index terms

Lazy functional programs behave differently from imperative programs and these differences extend to cache behaviour. We use hardware counters and a simple yet accurate execution cost model to analyse some large Haskell programs on the x86 architecture. The programs do not interact well with modern processors---L2 cache data miss stalls and branch misprediction stalls account for up to 60% and 32% of execution time respectively. Moreover, the program code exhibits little exploitable instruction- ...

Keywords: Glasgow Haskell Compiler, Haskell, branch misprediction, cache measurement, cache simulation, hardware counters

10 A survey of processors with explicit multithreading

Theo Ungerer, Borut Robič, Jurij Šilc

March 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 1

Full text available: pdf(920.16 KB) Additional Information: full citation, abstract, references, index terms

Hardware multithreading is becoming a generally applied technique in the next generation of microprocessors. Several multithreaded processors are announced by industry or already into production in the areas of high-performance microprocessors, media, and network processors.A multithreaded processor is able to pursue two or more threads of control in parallel within the processor pipeline. The contexts of two or more threads of control are often stored in separate on-chip register sets. Unused i ...

Keywords: Blocked multithreading, interleaved multithreading, simultaneous multithreading

11 Performance of a hardware-assisted real-time garbage collector

William J. Schmidt, Kelvin D. Nilsen

November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 , 28 Issue 11 , 5

Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Hardware-assisted real-time garbage collection offers high throughput and small worst-case bounds on the times required to allocate dynamic objects and to access the memory contained within previously allocated objects. Whether the proposed technology is cost

effective depends on various choices between configuration alternatives. This paper reports the performance of several different configurations of the hardware-assisted real-time garbage collection system subjected to several different ...

12 MEDEA workshop: Fresh Breeze: a multiprocessor chip architecture guided by modular programming principles

Jack B. Dennis

March 2003 **ACM SIGARCH Computer Architecture News**, Volume 31 Issue 1

Full text available: pdf(862.10 KB) Additional Information: full citation, abstract, references, index terms

It is well-known that multiprocessor systems are vastly more difficult to program than systems that support sequential programming models. In a 1998 paper[11] this author argued that six important principles for supporting modular software construction are often violated by the architectures proposed for multiprocessor computer systems. The Fresh Breeze project concerns the architecture and design of a multiprocessor chip that can achieve superior performance while honoring these six principles. ...

13 Stride prefetching by dynamically inspecting objects

Tatsushi Inagaki, Tamiya Onodera, Hideaki Komatsu, Toshio Nakatani

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available: pdf(168.64 KB) Additional Information: full citation, abstract, references, citings, index terms

Software prefetching is a promising technique to hide cache miss latencies, but it remains challenging to effectively prefetch pointer-based data structures because obtaining the memory address to be prefetched requires pointer dereferences. The recently proposed stride prefetching overcomes this problem, but it only exploits *inter-iteration* stride patterns and relies on an off-line profiling method.We propose a new algorithm for stride prefetching which is intended for use in a dynamic ...

Keywords: Java just-in-time compiler, object inspection, stride prefetching

14 Cycles to recycle: garbage collection to the IA-64

Richard L. Hudson, J. Elliot Moss, Sreenivas Subramoney, Weldon Washburn

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management**, Volume 36 Issue 1

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, citings, index terms

The IA-64, Intel's 64-bit instruction set architecture, exhibits a number of interesting architectural features. Here we consider those features as they relate to supporting garbage collection (GC). We aim to assist GC and compiler implementors by describing how one may exploit features of the IA-64. Along the way, we record some previously unpublished object scanning techniques, and offer novel ones for object allocation (suggesting some simple operating system support that would simplify it ...

15 Memory forwarding: enabling aggressive layout optimizations by guaranteeing the safety of data relocation

Chi-Keung Luk, Todd C. Mowry

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual internati nal symp sium n C mputer architecture**, Volume 27 Issue 2

Full text available: pdf(196.77 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

By optimizing data layout at run-time, we can potentially enhance the performance of

caches by actively creating spatial locality, facilitating prefetching, and avoiding cache conflicts and false sharing. Unfortunately, it is extremely difficult to guarantee that such optimizations are *safe* in practice on today's machines, since accurately updating *all* pointers to an object requires perfect alias information, which is well beyond the scope of the compiler for languages such as C. T ...

16 Sentinel scheduling for VLIW and superscalar processors

Scott A. Mahlke, William Y. Chen, Wen-mei W. Hwu, B. Ramakrishna Rau, Michael S. Schlansker

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

Speculative execution is an important source of parallelism for VLIW and superscalar processors. A serious challenge with compiler-controlled speculative execution is to accurately detect and report all program execution errors at the time of occurrence. In this paper, a set of architectural features and compile-time scheduling support referred to as sentinel scheduling is introduced. Sentinel scheduling provides an effective framework for compiler-controlled speculative ex ...

17 Performance: Method-level phase behavior in java workloads

Andy Georges, Dries Buytaert, Lieven Eeckhout, Koen De Bosschere

October 2004 **Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**

Full text available: pdf(695.63 KB) Additional Information: full citation, abstract, references, index terms

Java workloads are becoming more and more prominent on various computing devices. Understanding the behavior of a Java workload which includes the interaction between the application and the virtual machine (VM), is thus of primary importance during performance analysis and optimization. Moreover, as contemporary software projects are increasing in complexity, automatic performance analysis techniques are indispensable. This paper proposes an off-line method-level phase analysis approach for ...

18 Superscalar design: Three extensions to register integration

Vlad Petric, Anne Bracy, Amir Roth

November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, index terms
Publisher Site

Register integration (or just integration) is a register renaming discipline that implements instruction reuse via physical register sharing. Initially developed to perform squash reuse, the integration mechanism can exploit more reuse scenarios. Here, we describe three extensions to the original design that expand its applicability and boost its performance impact. First, we extend squash reuse to general reuse. Whereas squash reuse maintains the concept of an instruction instance "owning" its ...

19 The use of multithreading for exception handling

Craig B. Zilles, Joel S. Emer, Gurindar S. Sohi

November 1999 **Pr ceedings f the 32nd annual ACM/IEEE internati nal symp sium n Micr architecture**

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings, index terms
Publisher Site

Common hardware exceptions, when implemented by trapping, unnecessarily serialize program execution in dynamically scheduled superscalar processors. To avoid the consequences of trapping the main program thread, multithreaded CPUs can exploit control and data independence by executing the exception handler in a separate hardware context. The main thread doesn't squash instructions after the excepting instruction, conserving fetch bandwidth and allowing execution of instructions inde ...

20 EPIC compilation: Optimization for the Intel® Itanium® architecture register stack

Alex Settle, Daniel A. Connors, Gerolf Hoflehner, Dan Lavery

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available: pdf(906.60 KB) Publisher Site

Additional Information: full citation, abstract, references, citings, index terms

The Intel® Itanium® architecture contains a number of innovative compiler-controllable features designed to exploit instruction level parallelism. New code generation and optimization techniques are critical to the application of these features to improve processor performance. For instance, the Itanium® architecture provides a compiler-controllable virtual register stack to reduce the penalty of memory accesses associated with procedure calls. The Itanium® Register Stack Engine ...

Your search matched **69** of **1099723** documents.
A maximum of **500** results are displayed, **50** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enterin new one in the text box.

vliw<and>memory Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Feedback-directed memory disambiguation for embedded multimedi VLIW computing**
Jae-Woo Ahn; Soo-Mook Moon; Wonyong Sung;
Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Sympo on , Volume: 2 , 6-9 May 2001
Pages:461 - 464 vol. 2

[Abstract] [PDF Full-Text (340 KB)] **IEEE CNF**

---

2 **Design study of shared memory in VLIW video signal processors**
Wu, Z.; Wolf, W.;
Parallel Architectures and Compilation Techniques, 1998. Proceedings. 1998 International Conference on , 12-18 Oct. 1998
Pages:52 - 59

[Abstract] [PDF Full-Text (88 KB)] **IEEE CNF**

---

3 **Reducing cost and tolerating defects in page-based intelligent mem**
Oskin, M.; Keen, D.; Hensley, J.; Lita, L.-V.; Chong, F.T.;
Computer Design, 2000. Proceedings. 2000 International Conference on , 17-2 Sept. 2000
Pages:276 - 284

[Abstract] [PDF Full-Text (852 KB)] **IEEE CNF**

---

4 **A semi-f lded instructi n f rmat f r VLIW architecture**
Won-Kee Hong; Seung-Yup Lee; Shin-Dug Kim;
ASICs, 1999. AP-ASIC '99. The First IEEE Asia Pacific Conference on , 23-25 A 1999
Pages:95 - 98

[Abstract] [PDF Full-Text (388 KB)] IEEE CNF

---

5 **Expl iting ILP in page-based intelligent mem ry**
Oskin, M.; Hensley, J.; Keen, D.; Chong, F.T.; Farrens, M.; Chopra, A.;
Microarchitecture, 1999. MICRO-32. Proceedings. 32nd Annual International Symposium on , 16-18 Nov. 1999
Pages:208 - 218

[Abstract] [PDF Full-Text (288 KB)] IEEE CNF

---

6 **A microprocessor with a 128-bit CPU, ten floating-point MAC's, four floating-point dividers, and an MPEG-2 decoder**
Suzuoki, M.; Kutaragi, K.; Hiroi, T.; Magoshi, H.; Okamoto, S.; Oka, M.; Ohba Yamamoto, Y.; Furuhashi, M.; Tanaka, M.; Yutaka, T.; Okada, T.; Nagamatsu, Urakawa, Y.; Funyu, M.; Kunimatsu, A.; Goto, H.; Hashimoto, K.; Ide, N.; Murakami, H.; Ohtaguro, Y.; Aono, A.;
Solid-State Circuits, IEEE Journal of , Volume: 34 , Issue: 11 , Nov. 1999
Pages:1608 - 1618

[Abstract] [PDF Full-Text (640 KB)] IEEE JNL

---

7 **Modulo scheduling for a fully-distributed clustered VLIW architectur**
Sanchez, J.; Gonzalez, A.;
Microarchitecture, 2000. MICRO-33. Proceedings. 33rd Annual IEEE/ACM International Symposium on , 10-13 Dec. 2000
Pages:124 - 133

[Abstract] [PDF Full-Text (800 KB)] IEEE CNF

---

8 **Proceedings of the 25th Annual International Symposium on Microarchitecture. MICRO 25 (Cat. No.92TH0497-8)**

Microarchitecture, 1992. MICRO 25., Proceedings of the 25th Annual Internatio Symposium on , 1-4 Dec. 1992
[Abstract] [PDF Full-Text (388 KB)] IEEE CNF

---

9 **Automatic generation of architectural models for designing dedicted VLIW signal processors**
Menez, G.; Auguin, M.; Boeri, F.; Carriere, C.;
Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on , Volume: 5 , 23-26 March 1992
Pages:541 - 544 vol.5

[Abstract] [PDF Full-Text (304 KB)] IEEE CNF

---

10 **VLIW-in-the-large: a model for fine grain parallelism exploitation distributed memory multiprocessors**
Danelutto, M.; Vanneschi, M.;
Microprogramming and Microarchitecture. Micro 23. Proceedings of the 23rd A Workshop and Symposium., Workshop on , 27-29 Nov. 1990
Pages:7 - 16

[Abstract] [PDF Full-Text (812 KB)] IEEE CNF

---

11 **Scheduled dataflow: executi n paradigm, architecture, and perf rmance evaluati n**
Kavi, K.M.; Giorgi, R.; Arul, J.;
Computers, IEEE Transactions on , Volume: 50 , Issue: 8 , Aug. 2001
Pages:834 - 846

[Abstract] [PDF Full-Text (1088 KB)] IEEE JNL

---

12 **Multi-thread VLIW processor architecture for HDTV decoding**
Hansoo Kim; Woo-Seung Yang; Myoung-Cheol Shin; Seung-Jai Min; Seong-Ok Bae; In-Cheol Park;
Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000 , 21-24 May 2000
Pages:559 - 562

[Abstract] [PDF Full-Text (384 KB)] IEEE CNF

---

13 **Cache sensitive module scheduling**
Sanchez, F.J.; Gonzalez, A.;
Microarchitecture, 1997. Proceedings. Thirtieth Annual IEEE/ACM Internationa Symposium on , 1-3 Dec. 1997
Pages:338 - 348

[Abstract] [PDF Full-Text (1052 KB)] IEEE CNF

---

14 **Higher performance and lower power enhancements to VLIW architectures**
Gass, W.;
Signal Processing Systems, 2001 IEEE Workshop on , 26-28 Sept. 2001
Pages:157

[Abstract] [PDF Full-Text (45 KB)] IEEE CNF

---

15 **Automatic and efficient evaluation of memory hierarchies for embe systems**
Abraham, S.G.; Mahlke, S.A.;
Microarchitecture, 1999. MICRO-32. Proceedings. 32nd Annual International Symposium on , 16-18 Nov. 1999
Pages:114 - 125

[Abstract] [PDF Full-Text (156 KB)] IEEE CNF

---

16 **Parallelization resources of image processing algorithms and their mapping on a programmable parallel videosignal processor**
Pirsch, P.; Kneip, J.; Ronner, K.;
Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on , Volume: 1 , 28 April-3 May 1995
Pages:562 - 565 vol.1

[Abstract] [PDF Full-Text (372 KB)] IEEE CNF

---

17 **Proceedings of 27th Hawaii International Conference on System Sciences (HICSS-27)**

System Sciences, 1994. Vol. I: Architecture, Proceedings of the Twenty-Seven Hawaii Internation Conference on , Volume: 1 , 4-7 Jan. 1994
[Abstract] [PDF Full-Text (248 KB)] IEEE CNF

---

18 **A VLIW architecture based on shifting register files**
Ugurdag, H.F.; Papachristou, C.A.;
Microarchitecture, 1993. Proceedings of the 26th Annual International Sympos on , 1-3 Dec. 1993
Pages:263 - 268

[Abstract] [PDF Full-Text (556 KB)] IEEE CNF

---

19 **Pathlength reduction features in the PA-RISC architecture**
Lee, R.; Mahon, M.; Morris, D.;
Compcon Spring '92. Thirty-Seventh IEEE Computer Society International Conference, Digest of Papers. , 24-28 Feb. 1992
Pages:129 - 135

[Abstract] [PDF Full-Text (612 KB)] IEEE CNF

---

20 **A programmable instruction format extension to VLIW architecture**
De Gloria, A.; Faraboschi, P.;
CompEuro '92 . 'Computer Systems and Software Engineering',Proceedings. , May 1992
Pages:35 - 40

[Abstract] [PDF Full-Text (472 KB)] IEEE CNF

---

21 **GFLOPS: a general flexible linearly organized parallel structure for images**
Houzet, D.; Basille, J.-L.; Latil, J.-Y.;
Application Specific Array Processors, 1991. Proceedings of the International Conference on , 2-4 Sept. 1991
Pages:431 - 444

[Abstract] [PDF Full-Text (612 KB)] IEEE CNF

---

22 **Hardware implementation of a general multi-way jump mechanism**
Soo-Mook Moon; Carson, S.D.; Agrawala, A.K.;
Microprogramming and Microarchitecture. Micro 23. Proceedings of the 23rd A Workshop and Symposium., Workshop on , 27-29 Nov. 1990
Pages:38 - 45

[Abstract] [PDF Full-Text (600 KB)] IEEE CNF

---

23 **Reducing the number of instructions**
Gusev, M.; Misev, A.; Popovski, G.; Mitrevski, P.;
Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on , 13-16 June 2000
Pages:55 - 60

[Abstract] [PDF Full-Text (512 KB)] IEEE CNF

---

24 **Interaction between sub-w rd parallelism expl itati n and l w p w code transformati ns f r VLIW multi-media pr cess rs**
Masselos, K.; Catthoor, F.; Goutis, C.E.; DeMan, H.;
Low-Power Design, 1999. Proceedings. IEEE Alessandro Volta Memorial Works on , 4-5 March 1999
Pages:52 - 60

[Abstract] [PDF Full-Text (304 KB)] IEEE CNF

---

25 **Dynamic binary translation and optimization**
Ebcioglu, K.; Altman, E.; Gschwind, M.; Sathaye, S.;
Computers, IEEE Transactions on , Volume: 50 , Issue: 6 , June 2001
Pages:529 - 548

[Abstract] [PDF Full-Text (6164 KB)] IEEE JNL

---

26 **Cost-conscious strategies to increase performance of numerical programs on aggressive VLIW architectures**
Lopez, D.; Llosa, J.; Valero, M.; Ayguade, E.;
Computers, IEEE Transactions on , Volume: 50 , Issue: 10 , Oct. 2001
Pages:1033 - 1051

[Abstract] [PDF Full-Text (2488 KB)] IEEE JNL

---

27 **A design study of a 0.25-µm video signal processor**
Dutta, S.; O'Connor, K.J.; Wolf, W.; Wolfe, A.;
Circuits and Systems for Video Technology, IEEE Transactions on , Volume: 8 , Issue: 4 , Aug. 1998
Pages:501 - 519

[Abstract] [PDF Full-Text (512 KB)] IEEE JNL

---

28 **The impact of Mpact 2**
Purcell, S.;
Signal Processing Magazine, IEEE , Volume: 15 , Issue: 2 , March 1998
Pages:102 - 107

[Abstract] [PDF Full-Text (612 KB)] IEEE JNL

---

29 **Processing the new world of interactive media**
Rathman, S.; Slavenburg, G.;
Signal Processing Magazine, IEEE , Volume: 15 , Issue: 2 , March 1998
Pages:108 - 117

[Abstract] [PDF Full-Text (1020 KB)] IEEE JNL

---

30 **VLIW DSP vs. superscalar implementati n f a baseline 11.263 vide enc der**
Banerjee, S.; Sheikh, H.R.; John, L.K.; Evans, B.L.; Bovik, A.C.;
Signals, Systems and Computers, 2000. Conference Record of the Thirty-Four

Asilomar Conference on , Volume: 2 , 29 Oct.-1 Nov. 2000
Pages:1665 - 1669 vol.2

[Abstract] [PDF Full-Text (424 KB)] IEEE CNF

---

31 **Hipar-DSP-a scalable family of high performance DSP-cores**
Wittenburg, J.P.; Hinrichs, W.; Lieske, H.; Kloos, H.; Friebe, L.; Pirsch, P.;
ASIC/SOC Conference, 2000. Proceedings. 13th Annual IEEE International , 13 Sept. 2000
Pages:92 - 96

[Abstract] [PDF Full-Text (540 KB)] IEEE CNF

---

32 **Design-for-testability of the FLOVA**
Daehan Youn; Ohyoung Song; Hoon Chang;
ASICs, 2000. AP-ASIC 2000. Proceedings of the Second IEEE Asia Pacific Conference on , 28-30 Aug. 2000
Pages:319 - 322

[Abstract] [PDF Full-Text (340 KB)] IEEE CNF

---

33 **High-level synthesis of nonprogrammable hardware accelerators**
Schreiber, R.; Aditya, S.; Ramakrishna Rau, B.; Kathail, V.; Mahlke, S.; Abrah S.; Snider, G.;
Application-Specific Systems, Architectures, and Processors, 2000. Proceeding IEEE International Conference on , 10-12 July 2000
Pages:113 - 124

[Abstract] [PDF Full-Text (236 KB)] IEEE CNF

---

34 **Alternative architectures for video signal processing**
Wolf, W.;
VLSI, 2000. Proceedings. IEEE Computer Society Workshop on , 27-28 April 2
Pages:5 - 8

[Abstract] [PDF Full-Text (32 KB)] IEEE CNF

---

35 **Memory hierarchy design for Jetpipeline: to execute scalar and vect instructions in parallel**
Sasaki, T.; Nakaike, T.; Takano, K.; Katahira, M.; Kobayashi, H.; Nakamura, T
Parallel Algorithms/Architecture Synthesis, 1997. Proceedings. Second Aizu International Symposium , 17-21 March 1997
Pages:66 - 73

[Abstract] [PDF Full-Text (632 KB)] IEEE CNF

---

36 **Evaluation of a memory hierarchy for the MTS multithreaded pr ces**
Gallagher, W.L.; Chuan-Lin Wu;
Parallel and Distributed Systems, 1997. Proceedings., 1997 International Conference on , 10-13 Dec. 1997
Pages:346 - 351

[Abstract] [PDF Full-Text (500 KB)] IEEE CNF

---

37 **Evaluati n f scheduling techniques n a SPARC-based VLIW testbe**
Seongbae Park; SangMin Shim; Soo-Mook Moon;
Microarchitecture, 1997. Proceedings. Thirtieth Annual IEEE/ACM Internationa Symposium on , 1-3 Dec. 1997
Pages:104 - 113

[Abstract] [PDF Full-Text (1104 KB)] IEEE CNF

---

38 **Strategies in a cost-effective implementation of the PDC half-rate c for wireless communications**
Fettweis, G.; Shihua Wang; Kobayashi, S.; Kawasaki, T.; Fujimaki, S.; Muwafi Gupta, K.; Mitsutake, M.; Kameshima, Y.; Konoma, H.; Zingman, J.; Wu, E.; Mizushima, Y.; Nambu, Y.; Yamozoe, K.;
Vehicular Technology Conference, 1996. 'Mobile Technology for the Human Ra IEEE 46th , Volume: 1 , 28 April-1 May 1996
Pages:203 - 207 vol.1

[Abstract] [PDF Full-Text (428 KB)] IEEE CNF

---

39 **The Mpact media processor redefines the multimedia PC**
Foley, P.;
Compcon '96. 'Technologies for the Information Superhighway' Digest of Papers , 25-28 Feb. 1996
Pages:311 - 318

[Abstract] [PDF Full-Text (676 KB)] IEEE CNF

---

40 **IMAP-CE: a 51.2 GOPS video rate image processor with 128 VLIW processing elements**
Kyo, S.; Koga, T.; Okazaki, S.;
Image Processing, 2001. Proceedings. 2001 International Conference on , Volu 3 , 7-10 Oct. 2001
Pages:294 - 297 vol.3

[Abstract] [PDF Full-Text (328 KB)] IEEE CNF

---

41 **CALiBeR: a software pipelining algorithm for clustered embedded V processors**
Akturan, C.; Jacome, M.F.;
Computer Aided Design, 2001. ICCAD 2001. IEEE/ACM International Conferen on , 4-8 Nov. 2001
Pages:112 - 118

[Abstract] [PDF Full-Text (774 KB)] IEEE CNF

---

42 **Debug facilities in the TriMedia CPU64 architecture**
Vranken, H.;
Test Workshop 1999. Proceedings. European , 25-28 May 1999
Pages:76 - 81

[Abstract] [PDF Full-Text (96 KB)] IEEE CNF

---

43 **S ftware prefetching f r s ftware pipelined l ps**

Sanchez, F.J.; Gonzalez, A.;
System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on , Volume: 7 , 6-9 Jan. 1998
Pages:778 - 779 vol.7

[Abstract] [PDF Full-Text (32 KB)] IEEE CNF

---

44 **Improving instruction-level parallelism by loop unrolling and dynam memory disambiguation**
Davidson, J.W.; Jinturkar, S.;
Microarchitecture, 1995. Proceedings of the 28th Annual International Sympos on , 29 Nov.-1 Dec. 1995
Pages:125 - 132

[Abstract] [PDF Full-Text (712 KB)] IEEE CNF

---

45 **Ultra fine-grain template-driven synthesis**
Kolson, D.J.; Dutt, N.; Nicolau, A.;
VLSI Design, 1994., Proceedings of the Seventh International Conference on , Jan. 1994
Pages:25 - 28

[Abstract] [PDF Full-Text (316 KB)] IEEE CNF

---

46 **Speculative disambiguation: a compilation technique for dynamic memory disambiguation**
Huang, A.S.; Slavenburg, G.; Shen, J.P.;
Computer Architecture, 1994. Proceedings the 21st Annual International Symposium on , 18-21 April 1994
Pages:200 - 210

[Abstract] [PDF Full-Text (864 KB)] IEEE CNF

---

47 **A fine-grained MIMD architecture based upon register channels**
Gupta, R.;
Microprogramming and Microarchitecture. Micro 23. Proceedings of the 23rd A Workshop and Symposium., Workshop on , 27-29 Nov. 1990
Pages:28 - 37

[Abstract] [PDF Full-Text (736 KB)] IEEE CNF

---

48 **An investigation of static versus dynamic scheduling**
Love, C.E.; Jordan, H.F.;
Computer Architecture, 1990. Proceedings. 17th Annual International Sympos on , 28-31 May 1990
Pages:192 - 201

[Abstract] [PDF Full-Text (816 KB)] IEEE CNF

---

49 **The first MAJC micr pr cess r: a dual CPU system- n-a-chip**
Kowalczyk, A.; Adler, V.; Amir, C.; Chiu, F.; Choon Ping Chng; De Lange, W.J. Yuefei Ge; Ghosh, S.; Tan Canh Hoang; Baoqing Huang; Kant, S.; Kao, Y.S.; C Khieu; Kumar, S.; Lan Lee; Liebermensch, A.; Xin Liu; Malur, N.G.; Martin, A.

Ngo, H.; Sung-Hun Oh; Orginos, I.; Shih, L.; Sur, B.; Tremblay, M.; Tzeng, A. D.; Zambere, S.; Jin Zong;
Solid-State Circuits, IEEE Journal of , Volume: 36 , Issue: 11 , Nov. 2001
Pages:1609 - 1616

[Abstract] [PDF Full-Text (327 KB)] **IEEE JNL**

---

50 **Architecture and applications of the HiPAR video signal process r**
Ronner, K.; Kneip, J.;
Circuits and Systems for Video Technology, IEEE Transactions on , Volume: 6 , Issue: 1 , Feb. 1996
Pages:56 - 66

[Abstract] [PDF Full-Text (1332 KB)] **IEEE JNL**

---

1 2 Next

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top